프로젝트 최종 보고서

|  |
| --- |
| 1. 프로그램 개요   * 방 추천 프로그램 * 회원 로그인, 검색 필터, 방 검색, 검색 목록, 구매 요청, 방 등록, 방 삭제 등. * 기존 데이터 로드 및 저장   2. 전체 논리 구조도  3. 모듈 별 기능   * + 1. App        1. 파일 로드 – 데이터를 읽어와 메모리에 저장함        2. 고객 로그인 – 고객의 아이디, 비밀번호와 비교하여 개인 계정에 로그인 함        3. 관리자 로그인 – 관리자 아이디, 비밀번호와 비교하여 관리자 계정에 로그인 함        4. 로그아웃 – 현재 저장된 로그인 정보를 초기화함        5. 종료 – 현재까지의 정보를 저장하고 프로그램을 종료함     2. User        1. 구매자           1. 검색 필터 설정 – 좌표를 이용한 거리, 보증금 월세 매매가의 범위, 특정 검색어를 포함하는 필터를 설정할 수 있음.           2. 방 검색 – 필터에 설정된 정보가 일치하는 방 출력           3. 구매 신청 – 판매자에게 구매할 의도를 전달함           4. 최근 검색 내역 / 구매 신청 내역을 출력        2. 판매자           1. 방 출력 – 현재 판매중인 방을 모두 출력함           2. 방 등록 – 정보를 입력하고 신규로 방을 등록함           3. 방 삭제 – 계약이 완료된 방을 사용자가 등록한 매물 목록에서 삭제함        3. Room           1. 전세/매매 데이터를 저장        4. 프로세스 처리도 & 시나리오 흐름도     5. 모듈의 함수    a. App();  /\*======================================= App==============================================  #pragma once  #include "Seller.h"  #include "Buyer.h"  #include "RoomForRent.h"  #include "RoomForSale.h"  #include <vector>  using namespace std;  class App  {  public:  App(); //기본 생성자  ~App(); //소멸자  void run(); //프로그램 시행  private:  vector<User\*> users; //유저 벡터  vector<Room\*> rooms; //방 벡터  User\* loginedUser; //로그인 유저  vector<Room\*> reusltOfFilter1, resultOfFilter2; //조건1, 2에 대한 필터링을 수행한 임시 검색 결과  int searchAreaRadius; //사용자 중심으로 검색할 범위(거리)  bool isFilter1Skipped, istFilter2Skipped, isFilter3Skipped; //필터링을 수행할 것인지 저장하는 변수들  int minDeposit, maxDeposit; //필터링에 사용될 보증금의 범위를 저장하는 변수들  int minMonthlyPay, maxMonthlyPay; //필터링에 사용될 월세의 범위를 저장하는 변수들  int minCost, maxCost; //필터링에 사용될 매매가의 범위를 저장하는 변수들    vector<Room\*> getFilter1Result(); // 필터1 반환 함수  vector<Room\*> getFilter2Result(); // 필터2 반환 함수  vector<Room\*> getFilter3Result(); // 필터3 반환 함수  void loadUserFile(); //유저 데이터 로드  void loadRoomFile(); //방 데이터 로드  void saveFiles(); //파일 저장  void run(); //프로그램 시행  void printMenu(); //메뉴 출력  bool login(string userID, string password); //로그인  bool logout(); //로그아웃    void printRecentlySearcedRoomsOfLoginedUser(); //최근 검색 내역 출력  Room\* getRoomByID(int \_roomID); //ID를 이용해 Room 반환  bool printFilterConditions(); //현재 필터들의 설정을 출력  void searchRooms(); //방 검색  void registerNewRoom(); // 방 등록  void deleteRoom(); //방 삭제  };==================================================================================================\*/  b. User  /\*======================================= User==============================================  #pragma once  #include <string>  using namespace std;  class User  {  public:  User(); //기본 생성자  ~User(); //소멸자  virtual void printUserInfo() = 0; //정보를 출력하는 순수 가상함수  private:  int userType; //유저 타입  string id; //아이디  string password; //패스워드    };    =============================================================================================\*/  c. Buyer  /\*======================================= Buyer==============================================  #pragma once  #include "User.h"  #include <list>  class Buyer :  public User  {  public:  Buyer(); //기본 생성자  ~Buyer(); //소멸자  bool addRoomIDToRecentlySearched(int \_roomID); //최근 검색 내역에 추가  bool addRoomIDToRequestToBuy(int \_roomID); //구매 요청 내역에 추가  virtual void printUserInfo(); //구매자 정보 출력  private:  int xCoord, yCoord; //구매자의 좌표  list<int> recentlySearched; //최근 검색 내역 리스트  list<int> roomsRequestToBuy; //구매 요청 내역 리스트    };  ==================================================================================================\*/0  d. Seller  /\*======================================= Seller==============================================  #pragma once  #include "User.h"  #include <list>  class Seller :  public User  {  public:  Seller(); // 기본 생성자  ~Seller(); // 소멸자  void addRoomIDToRegisteredRoomList(int \_roomID); //iD를 이용해 방 추가  virtual void printUserInfo(); //판매자 정보 출력  private:  list<int> registerdRoomList; //등록한 방 리스트  };  ==================================================================================================\*/  e. Room  /\*======================================= Room==============================================  #pragma once  #include <list>  #include <string>  using namespace std;  class Room  {  public:  Room(); // 기본 생성자  ~Room(); // 소멸자  bool addUserInRequestToBuyList(string \_userID); // 유저ID를 이용해 구매 요청 리스트에 추가  virtual void printRoomInfo() = 0; // 정보를 출력하는 순수 가상 함수  private:  int roomType; // 방 타입  int roomID; // 방 ID  int sellerID; // 판매자 ID  string roomName; // 방 이름  int xCoord, yCoord; // 좌표  list<string> usersRequestToBuyList; // 구매 요청한 유저 ID 리스트    };  ==================================================================================================\*/  f. RoomForRent  /\*======================================= RoomForRent==============================================  #pragma once  #include "Room.h"  class RoomForRent :  public Room  {  public:  RoomForRent(); // 기본 생성자  ~RoomForRent(); // 소멸자  virtual void printRoomInfo(); // 방 정보 출력  private:  int deposit; // 보증금  int monthlyPay; // 월세    };  ==================================================================================================\*/  g. RoomForSale  /\*======================================= RoomForSale==============================================  #pragma once  #include "Room.h"  class RoomForSale :  public Room  {  public:  RoomForSale(); // 기본 생성자  ~RoomForSale(); // 소멸자  virtual void printRoomInfo(); // 방 정보 출력  private:  int cost; // 매매가    };  ==================================================================================================\*/  6. 설계 원칙  설계 요구 조건에 따라 충실히 설계한다.   1. 방 추천 프로그램을 구현하기 위한 클래스는 App, User, Seller, Buyer, Room, RoomForRent, RoomForSale 이다. 2. User 클래스를 상속받은 구매자 클래스 Seller와 판매자 클래스 Buyer가 있다. 3. Room 클래스를 상속받은 월세 클래스 RoomForRent와 매매 클래스 RoomForSale가 있다. 4. App은 User DB, Room DB를 가지고 있다. 5. User는 아이디, 비밀번호, 사용자 타입을 가지고 있다. 6. User를 상속받은 Seller는 자신이 등록한 매물 목록을 가지고 있다. 7. User를 상속받은 Buyer는 자신의 좌표, 최근 검색 내역, 보낸 구매요청 리스트를 가지고 있다. 8. App는 User와 Room 객체들을 총괄하여 관리하고 필요한 함수를 호출한다. 9. 프로그램 실행 시에 User DB, Room DB를 App 클래스의 각 컨테이너에 load한다. 10. 프로그램 종료 시에 갱신된 정보들을 파일에 다시 쓴다. 11. 구매자 기능(기본 정보, 검색 필터 설정, 방 검색, 구매 신청, 최근 검색 내역) 과 판매자 기능(기본 정보, 방 등록, 방 삭제)을 구현한다.   7. 개발도구  Microsoft Visual Studio 2013 C++  8. 사용 매뉴얼  **파일 로드**  **로그인**  **방 등록**  **필터 설정 및 검색**  **검색 / 구매 요청 내역**  **구매 요청**  **방 삭제**  **파일 저장**   1. 시작 메뉴       <프로그램 시작 화면>    <판매자 모드로 로그인 한 모습>    <구매자 모드로 로그인 한 모습>  <등록되지 않은 아이디를 입력한 모습>    <비밀번호를 잘못 입력한 모습>    <관리자 모드/구매자 모드로 구매자/관리자가 로그인 할 경우>   1. 구매자 로그인 / 판매자 로그인 : 입력 데이터와 저장된 ID, 비밀번호를 확인하여 개인 계정에 접속, 개인 정보와 메뉴 출력 (예외 처리 : 일치하는 ID가 없을 경우, 비밀번호가 일치하지 않을 경우, 이미 로그인 된 상태에서 재 로그인 할 경우, 지정된 숫자 외 다른 문자를 입력 받았을 경우, 고객 로그인에서 관리자 계정으로 로그인 시도 할 경우, 관리자 로그인에서 고객 계정으로 로그인 시도할 경우) 2. 종료 : 여태까지의 모든 데이터를 저장하고 프로그램을 종료   b. 방 등록    <판매자 메뉴에 접속했을 때의 인터페이스>    <판매자가 새 방을 등록하는 모습>    <잘못된 좌표를 입력한 경우>   1. 방 등록 여부 결정: 새 방을 등록할지 여부를 물음 2. 방의 종류를 설정: 월세인지 매매인지를 결정 3. 방의 기본 정보를 입력 : 이름, 좌표를 입력 받음. 월세의 경우 보증금과 월세를, 매매의 경우 매매가를 추가로 입력 받음. (예외처리 : 음수나 문자 입력 등 잘못된 입력을 받았을 경우)   **새 방**  **어플리케이션**  **전체 방 벡터**  **판매자**  **등록한 방 벡터**  **ID**   1. 방 등록 완료: 등록한 방은 어플리케이션의 전체 방 벡터로 들어가고 방의 ID는 판매자의 방 ID 벡터로 들어감   c. 방 삭제     1. 현재 등록된 방 출력: 유저가 등록한 방을 리스트에서 불러와 개수와 정보를 출력함 2. 삭제할 방의 아이디 입력: 사용자로부터 삭제할 방의 아이디를 입력받음 3. 방의 자세한 정보와 구매 요청한 사람 출력 4. 삭제 여부 확인: 마지막으로 삭제 여부를 확인   구매자  최근 검색 내역  구매 요청 내역  삭제할 방  삭제할 방  전체 방 벡터  삭제할 방  판매자 등록 방  삭제할 방   1. 삭제 : 방 삭제 시 구매 요청 리스트와 최근 검색 목록에 존재하는 방의 기록을 모두 삭제하고, 판매자가 등록한 방 벡터와 전체 방 벡터에서 방을 삭제   .  e. 필터 설정 및 검색   1. 필터 초기화 : 모든 필터는 “상관없음”으로 초기화 되어있음 2. 검색 반경 필터 설정 여부 : 로그인 된 구매자의 좌표로부터 일정 거리 내에 있는 방만 탐색할 것인지를 확인. (예외처리 : 음수 등 잘못된 입력) 3. 가격 범위 필터 설정 여부 : 특정 가격 범위 내에 있는 방만 탐색할 것인지를 확인 (예외처리 : 잘못된 입력, 최대값이 최소값보다 작을 경우) 4. 검색어 필터 설정 여부 : 특정 단어가 포함된 방만 탐색할 것인지를 확인   전체 방 벡터  필터 1  필터 2  필터 3  (최종 결과물)  구매자  최근 검색 내역   1. 검색 결과 : 필터에서 설정한 결과와 설정된 필터를 출력함 2. 자세한 정보 확인 : 방의 인덱스를 선택하여 자세한 정보를 확인 가능. 이때 최근 검색 내역에 방의 ID를 추가   f. 구매 요청     1. 검색 결과에서 자세한 정보 확인   고객  구매 요청 내역  구매하고자 하는 방  구매 신청 목록   1. 구매 요청에서 구매 선택 : 고객의 구매 요청 벡터에 해당 방의 ID가 추가되며, 해당 방의 구매 요청 벡터에 구매자의 ID가 삽입됨   G, 최근 검색 내역 / 구매 요청 내역     1. 최근 검색 내역 : 로그인한 구매자의 최근 검색 내역을 최고 5개 까지 출력함 2. 구매 요청 내역 : 로그인한 구매자의 구매 요청 내역을 출력함   구매자  최근 검색 내역  구매 요청 내역 |
|  |